Designing Technical Action Research and Generalizing from Real-World Cases

Roel Wieringa

Department of Electrical Engineering, Mathematics, and Computer Science,
University of Twente, The Netherlands
http://www.ewi.utwente.nl/~roelw

Abstract. This tutorial presents a sound methodology for technical action research, which consist of testing a new artifact by using it to solve a real problem. Such a test would be useless if we could not generalize from it, and the tutorial introduces architectural inference as a way of supporting generalizations by technical action research.

Keywords: Case studies, Action research, Technology validation, Generalization.

1 Introduction

A *case study* is an empirical investigation of an instance of a phenomenon in its natural context. For example, a researcher could investigate what the causes of delay are in ERP implementations by investigating one particular ERP implementation project. In some case studies, a researcher may investigate multiple cases, but because each individual case study requires considerable resources from the researcher, multiple case studies rarely involve more than four individual cases.

In an *observational case study,* the researcher tries to minimize his or her influence on the case. The researcher collects documents, interviews people, and observes their behavior without trying to change what happens in the case. For example, in an observational case study of a running ERP implementation project, the researcher will not intervene if he or she observes that the project will run over time. At the other extreme is the *action case study,* in which the aim of the researcher is to improve something in the case. For example, the researcher may join the ERP implementation project and help monitor its progress, signal problems, and provide knowledge about methods to speed up progress.

One particular kind of action case study is *technical action research* (TAR), which is an action case in which the researcher tests a technique by using it in practice [1]. For example, the researcher may have developed a new technique for effort estimation for ERP implementations. In a TAR project, the researcher joins an ERP implementation project and then uses this technique to estimate the effort of this project. The goal is both to help the project and to learn something about the technique.

J. Ralyté et al. (Eds.): CAiSE 2012, LNCS 7328, pp. 697–698, 2012.

In a TAR project, the researcher plays three roles, that must be kept separate:

- Technique developer. The researcher has developed some technique X.
- Client helper. The researcher helps some stakeholder, called the client, and uses X to do this.
- Empirical researcher. The researcher investigates the effects and utility of X in practice.

For example, the researcher (1) may have developed some new effort estimation technique, (2) use this technique in an ERP implementation project, and then (3) draws lessons learned from this use: Is it usable? Is it easier to use than alternative techniques? Does it give accurate results?

The answers to these questions would be of merely anecdotal value if the researcher could not generalize from them. In a TAR project, as in all other case studies, statistical inference is not applicable as a generalization technique, and the researcher must use case-based inference. In *case-based inference,* the researcher reasons from an observed case to an unobserved case by pointing out relevant similarities between the cases. For example, if an effort estimation technique has been effective in one project done for a large bank, it is plausible that it may be effective also in other projects done for large banks, because these other projects are probably similar to the current one.

But how similar must these projects be for this inference to be plausible? Not just any similarity is relevant. The relevance of the similarities must be based on a theory, and the inference is as plausible as this theory is. An effective way of case-based inference is to use *architectural theories,* which are theories that represent the case in terms of components with capabilities and interactions, and then identify mechanisms of interaction that are reproducible from case to case, and that explain the observations made in the case [2]. For example, suppose the effectiveness of an effort estimation technique in one case can be explained by the fact that the technique itself is based on a number of project factors that could be influenced by the project manger, then it is plausible that in all projects where these factors are present, and can be manipulated by the project manager, the same technique will be effective too. This generalization is based on the architecture of the projects (project factors, project manager) and mechanisms in the projects (influence of project manager on project factors).

References

1. Wieringa, R., Morali, A.: Technical Action Research as a Validation Method in Information Systems Design Science. In: Peffers, K., Rothenberger, M., Kuechler, B. (eds.) DESRIST 2012. LNCS, vol. 7286, pp. 220–238. Springer, Heidelberg (2012)
2. Wieringa, R., Daneva, M., Condori-Fernandez, N.: The structure of design theories, and an analysis of their use in software engineering experiments. In: International Symposium on Empirical Software Engineering and Measurement (ESEM), pp. 295–304. IEEE Computer Society (September 2011)